Proposition d'une architecture pour l'expérimentation de systèmes distribués extensibles

Michel Banâtre, Gilles Muller

Unité de recherche INRIA Lorraine, technopôle de Nancy-Brabois, 615 rue du jardin botanique, BP 101, 54600 VILLERS-LÈS-NANCY
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, domaine de Voluceau, Rocquencourt, BP 105, LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105 LE CHESNAY Cedex (France)

ISSN 0249-6399

Références

[Accetta *et al.* 86] M. Accetta, R. Baron, W. Bolosky, D. Golub, R. Rashid, A. Tevanian & M. Young. Mach: A New Kernel Foundation for Unix Development. *Proc. of Usenix 1986 Summer Conference*, pages 93–112, juillet 1986.

[Boisseau *et al.* 94] M. Boisseau, M. Demange & J.M. Munier. *Réseaux ATM*. EYROLLES, 1994.

[Fore Systems 94] Inc. Fore Systems. *200-Series ATM Adapter: Design and Architecture*. 174 Thorn Hill Road, Warrendale, PA 15086, 1994.

[Johnson & Zwaenepoel 91] J.H. Johnson & W. Zwaenepoel. The Peregrine high performance RPC system. Dept. of Computer Science COMP TR91-152, *Rice University*, 1991.

[Loepere 91] K. Loepere. *MACH 3 Kernel Principles*. Open Software Foundation, mars 1991.

[Loepere 93] K. Loepere. *OSF MACH 3 Kernel Final Draft Kernel Interfaces*. Open Software Foundation and Carnegie Mellon University, mai 1993.

[Maeda & Bershad 93] C. Maeda & B.N. Bershad. Protocol Service Decomposition for High-Performance Networking. OperatingSystem Review, éditeur, *Proc. of 14th ACM Symposium on Operating Systems Principles*, volume 27, pages 244–245, Asheville, NC, décembre 1993. ACM.

[Rozier *et al.* 88] M. Rozier, V. Abrossimov, F. Armand, I. Boule, M. Gien, M. Guillemont, F. Herrmann, P. Léonard, S. Langlois & W. Neuhauser. The Chorus Distributed Operating System. *Computing Systems*, 1(4):305–370, 1988.

[Thekkath & Levy 93] C.A. Thekkath & H.M. Levy. Low-Latency Communication on High-Speed Networks. *ACM Transactions on Computer Systems*, 11(2):179–203, mai 1993.

6 Synthèse

Dans ce document, nous avons donné les raisons qui nous ont incités à mettre en place une plate-forme d'expérimentation de systèmes distribués extensibles permettant de satisfaire les besoins posés par la recherche en systèmes et applications distribués : nécessité de disposer d'un outil de taille "réelle", support des nouvelles applications distribuées multimédia, préservation des développements existants et suivi de l'évolution technologique. Ensuite, nous avons justifié nos choix pour les différents éléments de la plate-forme :

- réseau ATM, lien de 100 Mbits en fibre optique ou de 155 Mbits sur paire torsadée,
- systèmes d'exploitation reposant sur la technologie micro-noyau Chorus à savoir Classix et Mix,
- stations de travail à base de PC/Pentium reposant sur les bus EISA et PCI.

La mise en œuvre de cette plate-forme aujourd'hui initialisée, sera effectuée progressivement, en fonction des différents travaux de portages de pilotes et de l'intégration de l'ATM dans le micro-noyau Chorus.

des pilotes. Actuellement, seule la société Fore satisfait ces deux contraintes. Une carte EISA est d'ores et déjà disponible, une carte PCI est annoncée pour le premier trimestre 1995.

5.5 Choix de l'interface Ethernet

De nombreuses cartes Ethernet sont disponibles pour PC, quelque soit le bus utilisé. Il est à noter que la majorité des cartes possède des tailles de transfert par mots de 8 ou 16 bits ; les mots de 32 bits ne sont pas supportés. De ce fait, il y a peu de différences en performance entre les cartes pour bus ISA et EISA. Des augmentations de performance sont prévisibles avec l'introduction du bus PCI, du fait de l'apparition d'une nouvelle génération de circuits intégrés contrôleurs.

Comme pour l'interface disque, les sources des cartes possédant une interface bus PCI ne sont pas encore disponibles dans le domaine public. Toutefois, contrairement aux disques, la performance de l'interface Ethernet n'est pas un paramètre critique de nos machines, étant donné que la majeure partie des communications à l'intérieur de la plate-forme sont effectuées *via* l'ATM. En conséquence, les diverses carte Ethernet SMC supportées par Chorus sont satisfaisantes.

Afin d'optimiser les communications avec l'environnement extérieur de la plate-forme, il est envisageable, dans un second temps, d'utiliser des cartes Ethernet reposant sur le bus PCI et de porter le pilote correspondant.

5.6 Choix de l'interface graphique, X-window

Il existe un choix entre de nombreuses cartes graphiques PCI reposant sur plusieurs types de circuits contrôleurs/accélérateurs. En ce qui concerne le serveur X, il existe un logiciel du domaine public, XFree, qui est compatible avec la plupart des versions d'Unix pour PCs (e.g., Mach/Bsd, Linux, OSF, SVR3, SVR4). Comme Chorus/Mix est compatible binaire avec SVR4, XFree peut s'exécuter sous Mix. XFree supporte également la majorité des contrôleurs graphiques, tout du moins ceux dont l'interface matérielle est publique et ne nécessite pas la signature d'un accord de confidentialité. En conséquence, les contraintes sur le choix d'un contrôleur graphique supportant X-window sont assez faibles.

dans les deux années à venir le bus EISA va disparaître au profit de PCI, une solution serait de s'équiper de PCs possédant deux bus EISA et PCI. Ceci nous permettrait d'utiliser le bus EISA dans une première phase et de migrer progressivement vers le bus PCI.

5.2 Choix du processeur

En ce qui concerne le processeur, il serait souhaitable que les performances des machines choisies soient toujours acceptables dans deux ou trois ans, malgré l'évolution de la technologie. D'où l'intérêt de choisir les processeurs de la famille Intel 0x86 actuellement les plus rapides, en l'occurrence les Pentium 66/90 Mhz. Toutefois, il est à noter que le noyau Chorus n'a pas encore été optimisé pour le Pentium et que pour l'instant Chorus est supporté seulement en mode compatible 486. Dans un second temps, il semble nécessaire de demander à Chorus Systèmes de réaliser un portage spécifique au Pentium afin de bénéficier des particularités de l'architecture de ce processeur.

5.3 Choix de l'interface disque

On rencontre essentiellement trois interfaces disques sur les PCs : ST506, IDE et SCSI. Historiquement la première, l'interface ST506 est aujourd'hui obsolète et n'est plus utilisée. L'interface IDE est utilisée dans les PCs de bas et moyenne gamme ; elle est relativement lente et la capacité des disques est assez limitée. Les interfaces SCSI et SCSI-2 sont aujourd'hui utilisées sur la majorité des stations de travail et PCs de haut de gamme. Cette interface est rapide et autorise la connexion de disques de grande capacité.

Afin d'avoir un accès rapide aux disques, la solution est d'utiliser l'interface SCSI. Dans ce but, il est nécessaire d'effectuer un portage des sources d'un pilote d'un contrôleur soit pour une carte EISA, soit pour une carte PCI. La meilleure solution serait d'utiliser le bus PCI. Cependant, les circuits contrôleurs SCSI (ex : NCR 53C810) pour ce bus sont trop récents et trop complexes pour que des pilotes fiables soient déjà disponibles dans le domaine public (par exemple, *via* le système d'exploitation Linux). En revanche, les sources de pilotes fiable pour des contrôleurs EISA, tel que l'Adaptec 174x, sont disponibles pour les systèmes Mach 3.0 et Linux.

Notre proposition est de débuter nos expérimentations sous Chorus/Mix avec des disques IDE, puis dans un second temps de porter soit un pilote SCSI/EISA-Adaptec, soit un pilote SCSI/PCI NCR 53C810 en fonction de la disponibilité des pilotes, afin de bénéficier de la performance de bus rapides.

5.4 Choix de l'interface ATM

Nos contraintes pour le choix d'une carte ATM sont (i) la disponibilité d'un contrôleur pour les bus EISA ou PCI, (ii) la disponibilité commerciale des sources

Le PC est une machine très ouverte, qui résulte d'une évolution de plus d'une dizaine d'années et qui comprend plusieurs génération de bus, de processeurs et une grande diversité de cartes d'interface. De ce fait, il est préférable de considérer le PC comme une architecture modulaire et d'examiner les différents choix pour chaque composant de l'architecture (disques, réseau, bus système). La difficulté de ce choix est de trouver des composants qui associent à la fois la performance et la disponibilité des sources du pilote.

La configuration du portage de référence PC de Chorus (386/486, bus ISA, disques IDE, réseau Ethernet contrôleurs de marque SMC) est inutilisable dans le cadre de la plate-forme, principalement pour des raisons de performance. En l'occurrence, le bus ISA est beaucoup trop lent pour une utilisation en tant que station de travail et est technologiquement dépassé. De plus, les disques au standard IDE sont également trop lents et de capacité insuffisante. Afin de construire une configuration PC performante, nous examinons maintenant les possibilités en matière de bus d'entrées-sorties, de processeur, de disque, de contrôleur Ethernet et de contrôleur graphique X-window.

5.1 Choix du bus d'entrées-sorties

Historiquement, plusieurs bus d'entrées-sorties ont été proposés pour le PC : ISA, EISA, MCA, VESA-Local-Bus, PCI-Local-Bus. Nous les comparons maintenant :

- ISA : ce bus est peu cher, lent (8 Mhz) et très bien standardisé. La taille des échanges est de 8 ou 16 bits. De nombreuses cartes sont disponibles,
- EISA : ce bus est relativement cher, rapide, mais assez peu diffusé à cause de son prix. La taille des échanges est de 8, 16 ou 32 bits,
- MCA : ce bus est d'origine IBM. Il est rapide, mais assez peu utilisé en dehors des machines IBM et BULL. La taille des échanges est de 8, 16 ou 32 bits,
- VESA : ce bus est une extension du bus ISA. Il est peu cher, rapide, mais pas très bien standardisé car non indépendant du processeur,
- PCI : ce bus est très rapide, mais encore cher. Il est indépendant du processeur et très bien standardisé. Sa mise en œuvre repose sur des circuits spécifiques et ne nécessite pas de circuits additionnels ("glue"). En conséquence, le prix des cartes PCI devrait baisser assez rapidement. La taille des échanges est de 8, 16, 32 ou 64 bits (Pentium uniquement). PCI est reconnu comme le bus PC des années à venir.

L'ATM nous impose de choisir le bus EISA, car la seule carte disponible actuellement repose sur l'utilisation de ce bus. En revanche, en ce qui concerne les performances, la configuration PC idéale serait basée sur le bus PCI, notamment pour le contrôleur de disque et l'interface graphique. Comme il est à peu près certain que

5 Choix des stations de travail

Les contraintes de choix des stations de travail sont de deux sortes : caractéristiques matérielles propres à la machine et disponibilité de l'environnement logiciel ou matériel nécessaire à la mise en œuvre de la plate-forme. En ce qui concerne les caractéristiques de la machine, nous devons prendre en compte les paramètres suivants :

- type du processeur (RISC ou CISC),
- nombre de processeurs (mono-processeur ou multiprocesseur),
- performance globale de l'architecture (cache, débit mémoire, débit des entrées/sorties),
- prix.

Les contraintes sur l'environnement logiciel et matériel sont les suivantes :

- disponibilité d'un portage de Chorus/Mix ou de Chorus/ClassiX,
- disponibilité d'une carte d'interface ATM.

Actuellement, la contrainte la plus forte repose sur la disponibilité de Chorus. La seule machine de référence, pour la quelle Chorus Système délivre et maintient les sources, est un PC-386/486, bus d'entrées/sortie ISA, disque IDE, interface réseau Ethernet "SMC Elite 16+". Si l'on désire effectuer un portage vers une autre machine, il est nécessaire de réaliser les travaux suivants :

- porter les modules sources de Chorus dépendants du type du processeur (MMU, registres, instructions spécifiques, …),
- écrire ou adapter des pilotes existants pour la configuration matérielle choisie.

Les modules de Chorus dépendants du processeur sont identiques pour toutes les machines construites à partir du même processeur. Ces modules peuvent être généralement fournies par Chorus Système grâce à l'existence de nombreux portages du micro-noyau. En revanche, les pilotes sont spécifiques à chaque architecture de machine et ne peuvent être réutilisés sans modifications.

L'écriture de pilotes à partir des spécifications du matériel (notices des circuits, schémas de l'architecture) est un travail très complexe et technique qui nécessite au minimum que les spécifications de l'architecture soient disponibles. L'adaptation de pilotes existants pour le même matériel, mais pour un autre système d'exploitation est plus aisée à réaliser. Dans le cas de pilotes écris pour le système SVR4, la réutilisation par Chorus est presque immédiate (cf. paragraphe 5.0.5). Toutefois, il est nécessaire que les sources des pilotes soient disponibles sans contraintes de confidentialité. La seule architecture de machine qui satisfasse actuellement cette condition est le PC.

4.7 Nos choix en matière de système d'exploitation

Les paragraphes précédents montrent que l'offre de Chorus Systèmes est la plus diversifiée et la plus apte à satisfaire nos besoins en matière de système d'exploitation *via* la combinaison des quatre solutions : simulateur, micro-noyau Chorus seul, Chorus/Mix et Chorus/ClassiX. En conséquence, nous choisissons Chorus comme base système de notre plate-forme.

Toutefois, dans des expérimentations qui nécessitent l'utilisation d'un multiprocesseur à mémoire partagée, tel que le Corollary, OSF/1 reste le système le mieux approprié, notamment en raison de ses performances sur ce type d'architectures. Il est donc utile d'assurer une veille technologique sur OSF/1 via l'acquisition régulière des nouvelles versions de ce système, sachant que nous avons déjà l'expérience de ce système via nos projets de recherche passés.

4.6.2. Chorus/Mix V.4

Chorus/Mix V.4 est une version modulaire de SVR4, conçue à partir de ce dernier, et qui est construite sur la version 4 du micro-noyau Chorus. Mix est un Unix possédant des extensions temps-réel. En ce qui concerne les applications, Mix est compatible binaire avec SVR4 et supporte également la réutilisation de code source noyau, tels que les pilotes d'entrées/sorties, respectant les spécifications DDI/DKI. Mix est un produit livré sous forme de sources dont l'achat requiert l'acquisition d'une licence SVR4 auprès de Novell (ex-USL).

Les avantages de Chorus/Mix sont multiples dans le contexte qui nous préoccupe :

- il nous permet d'utiliser le même système d'exploitation pour nos expérimentations (e.g., micro-noyau ou Unix) et nos besoins quotidiens,
- le système est modulaire ce qui nous permet d'étendre ou modifier ses fonctionnalités plus aisément que dans le cas d'un système monolithique,
- la communication entre les différents modules est effectuée par messages ce qui permettrait de distribuer la mise en œuvre de Mix,
- la compatibilité avec SVR4 permet de réutiliser les pilotes existants pour une machine donnée et d'éviter la ré-écriture de ceux-ci lors du portage de Mix sur cette machine.

4.6.3. Chorus/ClassiX

Pour certaines expérimentations, telles que la construction de systèmes spécialisés, il n'est pas nécessaire voire même désavantageux que l'expérimentation soit réalisée au dessus d'un système Unix. Dans ces situations, il est préférable d'utiliser le micro-noyau Chorus seul, mais avec toutefois la possibilité de disposer d'un environnement de mise au point minimal.

Chorus/ClassiX a été conçu dans ce but. C'est un mini-système d'exploitation qui assure les fonctions d'accès aux fichiers distants via un serveur NFS et de mise au point déportée à partir d'une station SunOs ou Chorus/Mix. Chorus/ClassiX est un produit très intéressant lorsque l'on utilise le micro-noyau Chorus sur des machines spécifiques, pour lesquelles on ne désire pas porter l'intégralité des pilotes (écran, disques, etc). Chorus/ClassiX est construit à partir du micro-noyau Chorus version 4 et d'un sous-ensemble du système Unix Bsd.

Outre les fonctionnalités précédentes, Chorus/ClassiX possède deux avantages principaux : (i) sa mise en œuvre ne nécessite que le portage d'un pilote Ethernet avec de plus un gestionnaire réseau identique à celui de Mix ; (ii) son acquisition ne requiers, aujourd'hui que la possession d'une licence BSD sachant qu'à terme Chorus/ClassiX sera libre de toute licence.

- l'interface du micro-noyau a évolué pour optimiser l'exécution du serveur OSF/1 et n'est plus compatible avec celle du CMU,
- le test et l'exécution d'autres serveurs demandent des privilèges Unix de type administrateur.

L'avantage principal de Norma-OSF/1 est l'existence de nombreux portages publics pour lesquels l'utilisation des sources ne nécessitent pas d'autre condition d'accès que la licence OSF/1. Parmi les machines de référence, on peut citer le PC et un certain nombre de multiprocesseurs à mémoire partagée : Corollary, Sequent. Une conséquence de l'utilisation de machines de référence multiprocesseurs est qu'OSF/1 a été optimisé par étude du placement des verrous d'exclusion mutuelle dans les divers modules du système*. De ce fait, OSF/1 est actuellement un des systèmes Unix les mieux optimisés pour ce genre d'architecture. Un autre avantage d'OSF/1 est sa commercialisation comme système natif par plusieurs constructeurs de machines dont Dec sur la série Alpha-AXP et Intel sur le multiprocesseur Paragon.

4.6 Offre de Chorus Systèmes

Au contraire d'OSF, la politique principale de Chorus Systèmes n'est pas de vendre une version Unix "clé-en-main", mais de la technologie micro-noyau *via* le micro-noyau lui-même (Versions 3 et 4) ou son simulateur. Chorus propose toutefois le code source d'une version modulaire d'Unix SVR4, appelée Chorus/MIX, qui permet de démontrer les possibilités du micro-noyau comme base de mise œuvre du système Unix. Il existe également une autre version d'Unix, appelée Chorus/Fusion, qui est compatible binaire avec le système de la société SCO. Cependant, cette version n'est distribuée que sous forme de produit binaire et ne peut donc être exploitée dans le cadre de développements de recherche. Nous décrivons maintenant plus précisément les produits qui nous intéressent.

4.6.1. Simulateur Chorus V3

Le simulateur Chorus est un programme s'exécutant sur des stations Sun avec le système SunOS qui émule le fonctionnement du micro-noyau Chorus version 3, aux performances près. Toutes les primitives du micro-noyau sont disponibles, à l'exception des routines dépendantes du matériel. Le simulateur peut fonctionner en environnement réparti et reproduire le comportement d'un réseau local de stations exécutant le micro-noyau Chorus. En conséquence, le simulateur est un excellent outil de test et de mise au point des applications qui ne sont pas dépendantes du matériel ou des performances. On peut noter à ce propos que le simulateur a servi à la mise au point d'une partie des fonctions du micro-noyau.

---

* Ce travail a été réalisé en grande partie par l'institut de recherche de l'OSF localisé à Grenoble.

à la suite de sa mise au point en tant que tâche système, et qui s'exécute dans l'espace d'adressage et de protection du noyau. Le relâchement de la protection offre l'avantage d'optimiser la communication entre serveurs superviseurs par suppression des changements de contexte et d'activité, et par passage des paramètres sur la pile.

4.3.3. Gestion des pilotes d'entrées-sorties

Les pilotes d'entrées-sorties représentent un point clé d'un système d'exploitation. Ils déterminent l'environnement extérieur accessible par le système et les utilisateurs. Comme les pilotes dépendent du matériel qui est spécifique à une machine donnée, leurs mises en œuvre constituent généralement la plus grande part de travail dans le portage d'un système d'exploitation.

Dans les micro-noyaux Mach et Norma, les pilotes sont intégrés au noyau. Leur interface et leur structure sont fixes. Les pilotes sont en fait très similaires à ceux d'Unix dont ils ont été hérités. En revanche, dans le micro-noyau Chorus, aucune structure de pilote n'est imposée. Tout acteur superviseur peut accéder aux contrôleurs d'entrées-sorties et remplir la fonction de pilote. On peut toutefois noter que la version d'Unix SVR4 proposée par Chorus, appelée MIX V.4 (cf paragraphe 5.0.5), offre un environnement d'émulation du noyau Unix qui permet la réutilisation de code système SVR4 tels que les pilotes d'entrées-sorties.

4.4 Offre de l'université de Carnegie Mellon (CMU)

Le CMU est à l'origine du projet de recherche Mach et distribue aujourd'hui des versions universitaires. Les conditions de distribution pour les universités sont les suivantes : le micro-noyau Mach seul est gratuit. En revanche, le serveur Unix BSD qui est nécessaire au fonctionnement du système sur un réseau, nécessite la possession d'une licence BSD.

Le projet de recherche Mach est maintenant terminé. Les seuls développements qui ont encore lieu sont relatifs à la maintenance du micro-noyau et à celle du serveur. De plus, il n'est pas du tout certain qu'un support existera encore dans un ou deux ans. Pour cette dernière raison, indépendamment de toute considération technologique, il semble risqué de choisir cette version de Mach comme système de base de la plate-forme.

4.5 Offre de l'OSF

L'offre de l'OSF en matière de système d'exploitation concerne principalement la fourniture d'OSF/1 qui est une version d'Unix. Comme l'OSF est une société commerciale, la disponibilité OSF/1 requiert l'achat d'une licence. Le micro-noyau Norma n'existe que dans le cadre d'OSF/1, ce qui a des conséquences négatives :

- il est très difficile de faire s'exécuter Norma sans le serveur OSF/1,

ciser que dans Mach, les communications noyau par messages sont locales à une machine. L'envoi d'un message sur une machine distante nécessite par conséquent l'utilisation d'un serveur relai qui gère les transmissions sur le réseau et masque la distribution au moyen d'un port local, cache du port distant. Dans la version standard distribuée par le CMU, le serveur qui implémente cette fonction (le *netmsgserver)* utilise le protocole TCP/IP pour assurer la fiabilité des transmissions réseau. Comme TCP/IP est lui-même mis en œuvre, en standard, par le serveur Unix, l'envoi d'un message à distance nécessite plusieurs échanges de messages locaux et est de ce fait assez inefficace.

Norma se distingue de la version universitaire de Mach par une intégration de la distribution au sein du noyau. Cette version de Mach a été initialement développée pour des multi-processeurs faiblement couplés tel que la Paragon. Il en résulte que ce système n'a pas été conçu pour gérer un réseau de machines indépendantes et que l'arrêt et le redémarrage d'une des machines sont impossibles sans relancer l'ensemble des machines du réseau. L'emploi de Norma dans un système distribué réel est de ce fait limité. Toutefois, cette limitation devrait être disparaître dans les versions futures de ce noyau.

Dans Chorus, un port est désigné par un identificateur global unique (UID, *Unique IDentifier*). Pour transmettre le nom d'un port à une autre tâche, il suffit d'émettre l'UID du port. La ressource port peut être *migrée* d'une tâche vers une autre à l'aide d'un appel noyau spécifique. L'appel de procédure à distance est fourni par le noyau Chorus et possède la sémantique au plus une fois. La gestion de la distribution et la localisation des ports sont complètement intégrés dans le noyau et ne nécessitent pas de serveur Unix. Le noyau fournit également la notion de groupe de ports vers lequel un message peut être diffusé.

4.3.2. Gestion des tâches et activités

Dans les micro-noyaux, la tâche réalise l'abstraction de protection et de désignation des entités, en définissant un espace d'adressage virtuel privé à la tâche. La gestion des tâches et des activités reposent sur un ensemble de fonctions de base similaires dans Mach, Norma et Chorus. Ces trois micro-noyaux peuvent s'exécuter sur des machines mono-processeur ou multi-processeurs. Toutefois, il est à noter que Norma et Chorus possèdent des extensions orientées temps réel.

Lorsque le système est mis en œuvre de manière modulaire en un ensemble de serveurs, la communication entre tâches engendre une dégradation des performances, notamment dû aux changement de contextes. Afin de concilier performance et modularité, le micro-noyau Chorus autorise l'exécution d'une tâche avec trois niveaux de privilèges : utilisateur, système et superviseur. Une tâche *utilisateur* représente la notion standard de tâche tel qu'on la trouve dans Mach et Norma. Une tâche *système* est une tâche utilisateur qui possède le droit d'exécuter des primitives "sensibles" du noyau tel que la connexion d'interruptions ou de nouveaux appels systèmes. Une tâche *superviseur* est une tâche dont le code est considéré comme éprouvé,

Dans le cadre de nos projets antérieurs, nous avons acquis, via Mach 3.0, une expérience dans l'utilisation des micro-noyaux pour le développement de systèmes et applications distribués. Aujourd'hui, l'investissement important que représente la plate-forme a été pour nous l'occasion de reconsidérer ce choix. Nous comparons maintenant les fonctionnalités de Mach, Norma et Chorus.

4.3 Comparaison de Mach 3.0, de Norma et de Chorus

S'il est reconnu que les micro-noyaux Chorus, Mach 3.0* et Norma sont assez semblables dans leurs fonctionnalités, ils possèdent cependant des différences significatives dans la gestion des tâches, des communications par messages et la gestion des entrées/sorties.

4.3.1. Gestion des communications entre processus (IPC)

Les primitives de communication par messages sont *envoyer* et *recevoir*. Dans Mach, les messages sont typés. Le message est constitué d'une suite de couples (type, données). A chaque envoi et réception de message, le noyau interprète les données en fonction de leur type. Ceci est particulièrement intéressant dans un système distribué hétérogène, car il est alors possible de prendre en compte les différences de représentation de données entre sites. Dans Norma**, qui est issu de Mach, le typage des ports a été supprimé pour des raisons de performance. Dans Chorus, les messages ne sont pas typés. Un message est composé d'une chaîne de taille variable contenant les données à transmettre, et d'une chaîne de taille fixe, appelée annexe, qui contient des informations de contrôle.

L'échange de message se fait via l'intermédiaire de structures particulières, les *ports*. Les ports sont des ressources de communication point-à-point détenues par une tâche. La communication est bidirectionnelle dans Chorus et unidirectionnelle dans Mach. A chaque port est associée une file de messages que les activités consomment. Seule la tâche qui possède un port peut recevoir des messages sur celui-ci.

Dans Mach, les noms des ports ne sont valides qu'au sein du contexte d'une tâche. Des droits d'émission et de réception sont associés aux ports. Le droit d'émission sur une porte peut être partagé par plusieurs tâches. En revanche, le droit de réception est attribué à une tâche et une seule. Les droits d'émission et de réception peuvent être transmis d'une tâche à une autre au moyen d'un message. Le typage des messages permet au noyau de retrouver le droit et de réaliser la transmission effective de celui-ci. Aux droits d'émission et de réception, Mach ajoute également le droit d'*émettre une fois*. Dans ce cas, le droit est détruit après émission. Ce mécanisme est utilisé pour mettre en œuvre l'appel de procédure à distance. Il est important de pré-

---

* Les versions de Mach antérieures à la version 3.0 ne possèdent pas réellement une structure de micro-noyau.

** Ceci est valide uniquement à partir de la version 16 de Norma.

4.2 Support des expérimentations de recherche

Les besoins en système d'exploitation pour le support des expérimentations de recherche sont assez différents de ceux de l'utilisation quotidienne. Généralement, les activités de recherche dans le domaine des systèmes d'exploitation consiste à expérimenter de nouveaux concepts, ce qui nécessite la modification ou l'extension d'un noyau de système. Cependant, il est rare que l'introduction d'une nouveau concept remette en cause l'ensemble des fonctions des systèmes existants. Le plus souvent, nous avons seulement besoin de modifier une seule des fonctions de base telles que la gestion mémoire, l'ordonnancement des processus ou la gestion de fichiers. De telles expérimentations sont simplifiées lorsque le système servant de base aux expérimentations est modulaire, ce qui permet de modifier une fonction sans avoir à intervenir dans l'ensemble du code du système.

Les systèmes d'exploitation ouverts reposants sur la technologie micro-noyau répondent à ce besoin. En effet, dans un système d'exploitation traditionnel, tel qu'UNIX, le noyau et les différents services sont intégrés en une seule entité (même espace d'adressage), d'où le nom de système monolithique. Au contraire, un système d'exploitation ouvert est formé de plusieurs entités distinctes, les serveurs, qui communiquent au moyen de messages par l'intermédiaire d'un micro-noyau. Une fonction système (ex: mémoire virtuelle, système de fichier) est alors mise en œuvre par un serveur ou un ensemble de serveurs qui s'exécutent chacun dans un espace d'adressage privé : la *tâche*, suivant la terminologie Mach, et *acteur*, suivant la terminologie Chorus. Le micro-noyau lui-même peut être considéré comme un serveur particulier, fournissant la communication par messages, la gestion de processus (tâche et activité "thread"), la gestion mémoire vive et l'interface avec le matériel (ex: processeur, unité de gestion mémoire).

Les avantages de la technologie micro-noyau sont aujourd'hui largement reconnus. Ils découlent directement du modèle de communication par messages et de la structuration du système d'exploitation en un ensemble de serveurs. Nous les énumérons succinctement ci-dessous :

- extension naturelle vers la distribution,
- construction de systèmes modulaires ou spécialisés,
- spécialisation de la gestion mémoire (mémoire virtuelle distribuée),
- fiabilité et sécurisation du système d'exploitation.

Parmi les principaux micro-noyaux développés au cours des années 80, seuls Chorus [Rozier *et al.* 88] et Mach [Accetta *et al.* 86][Loepere 91][Loepere 93] sont actuellement industrialisés. Il est à noter que la version de Mach industrialisée par l'OSF (Open Software Foundation) sous le nom de Norma diffère de la version universitaire du CMU.

Il est à noter que X-window et les commandes d'exécution à distance reposent sur le protocole TCP/IP. La connexion physique à la passerelle peut être réalisée soit par Ethernet, soit à travers le réseau ATM par émulation du protocole IP.

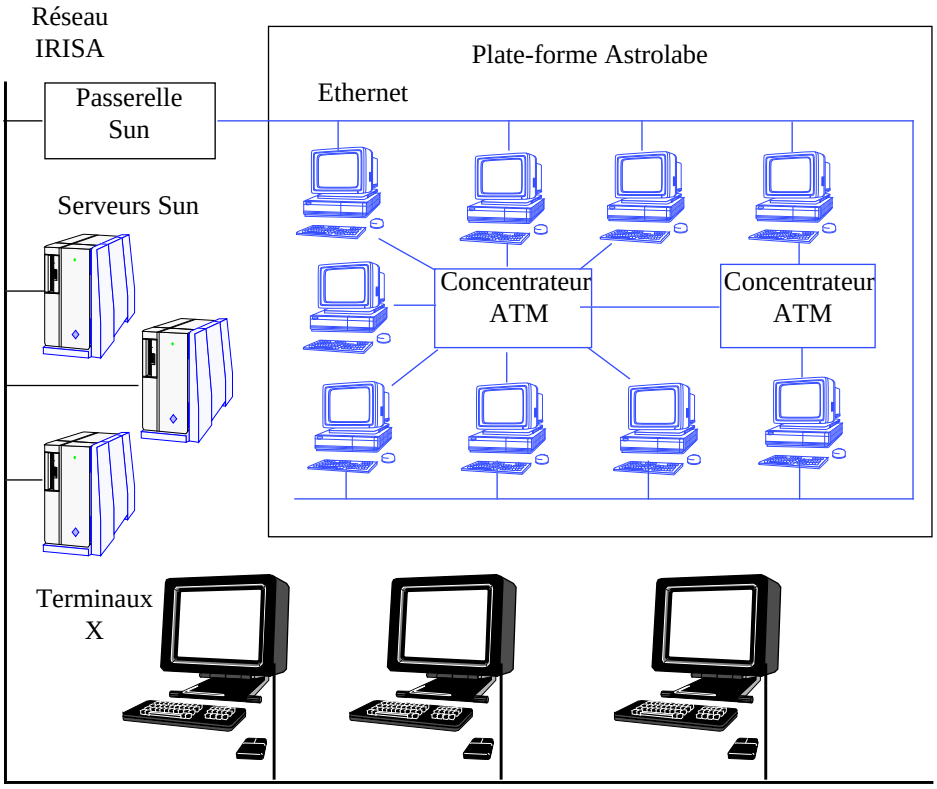

Figure 5 : Connexion entre la plate-forme et le réseau de l'IRISA

Outre l'accès aux serveurs de calcul, d'autres ressources communes du laboratoire telles que les serveurs de fichiers doivent également être accessibles. A l'heure actuelle, la fonction d'accès au fichiers distants est réalisée à l'Irisa au moyen de NFS. Il est donc nécessaire que la version d'Unix retenue supporte ce système de fichier.

Les divers besoins pour l'utilisation en mode "quotidien" peuvent être résumés par les critères suivants :

- nécessité d'un système Unix
- support du système de fichiers distribués NFS
- support de X-window

4 Aspects relatifs au système d'exploitation

Nos besoins en système d'exploitation sont de deux ordres : utilisation quotidienne et support des expérimentations de recherche en systèmes et applications distribuées. L'utilisation quotidienne relève principalement de l'édition de textes, du développement de programmes (écriture et compilation) et d'échanges d'informations avec la communauté scientifique nationale et internationale (e.g., Mail, News, WWW). L'expérimentation relève surtout de la conception, du test et de la mise au point des développements de recherche.

Les besoins de ces deux types d'activités sont très différents et ne nécessitent pas forcément le même système d'exploitation. Ceci nous amène à considérer deux modes d'utilisation ("quotidien" et "expérimentation") de nos machines avec un ou plusieurs systèmes d'exploitation associés à chaque activité. Le passage d'une activité à l'autre s'effectuant par redémarrage complet de la machine.

4.1 Utilisation quotidienne

En utilisation quotidienne, le nombre d'outils nécessaires est très important. Nous pouvons citer entre autre les éditeurs de documents (dessins et textes), les compilateurs et logiciels de gestion du développement de programmes, jusqu'aux outils de communication (Mail, News). Pour pouvoir fonctionner, ces outils requièrent généralement la présence d'un système d'exploitation de type Unix, permettant de travailler dans un environnement distribué.

Plus précisément, nous pouvons classer les outils en deux groupes : les logiciels dont les sources sont disponibles et qui sont conçus pour être portables (ex : logiciels du domaine public) et ceux dont on ne possède que le binaire (ex : FrameMaker) ou dont l'exécution ailleurs que les stations Suns du réseau de l'Irisa demande une maintenance système imporante (ex : News, Mail). L'utilisation des logiciels du deuxième groupe suppose que les utilisateurs puissent se connecter, depuis la plate-forme, à un ensemble de serveurs de calcul (cf figure 5). L'accès à ces serveurs étant réalisé via une exécution distante (ex : rlogin, rsh).

Pour ce qui est de l'interface homme-machine, il est souhaitable qu'elle soit indépendante de la machine utilisée (locale ou distante). Cette propriété peut être mise en œuvre au moyen du protocole X-window qui distingue les notions de machine client, sur laquelle s'exécute l'application, et de serveur d'écran (poste de travail) qui réalise les fonctions d'affichages graphiques et de saisie (souris/clavier). L'application client et le serveur peuvent soit s'exécuter sur la même machine, soit être physiquement distribués. Trois types de postes de travail peuvent être utilisés : les stations de la plate-forme, les Suns de l'Irisa et des terminaux X qui sont des machines dédiées à la fonction d'affichage et n'offrent pas de ressource calcul à l'utilisateur (cf figure 5).

SPECint comme unité de mesure, avec toutes les réserves concernant ce benchmark, nous obtenons un seuil de performance de 60 SPECint pour environ 50 Mbits/s.

3.7 Synthèse du cahier des charges pour le réseau

Dans ce paragraphe, nous synthétisons nos besoins en matière de réseau pour la réalisation de notre plate-forme. Notre objectif est de disposer dans les deux ans à venir d'une architecture composée environ d'une trentaine de machines interconnectées par un réseau ATM.

Afin de minimiser le temps de latence, il semble intéressant de limiter à deux le nombre maximum de concentrateurs traversés pour communiquer entre deux machines. Ceci permet de construire la plate-forme autour de trois concentrateurs en connectant une dizaine de machines à chacun d'eux. En cas de défaillance d'un concentrateur, la puissance globale de la plate-forme serait alors réduite d'un tiers. En considérant que deux voies sont nécessaires pour la connexion d'un concentrateur à ses deux voisins et qu'il est utile de réserver quatre voies pour les extensions futures, un concentrateur doit disposer d'au moins 16 voies d'accès.

En ce qui concerne les liaisons entre les machines et les concentrateurs, un débit de 100 Mbits/s ou 155 Mbits/s est nécessaire. En revanche, il est souhaitable que le débit des liaisons entre concentrateurs soit plus élevé, soit par réplication de liaisons 100/155 Mbits/s, soit par utilisation de liaisons de 600 Mbits/s.

Afin de résoudre les problèmes d'hétérogénéité, il est nécessaire que des machines (cartes d'interfaces) ou concentrateurs de sources diverses puissent être connectées à la plate-forme. Ceci suppose que les normes actuelles (UNI V3.0) et futures de l'ATM Forum soient supportées par les concentrateurs et les logiciels de gestion des cartes contrôleurs ATM. Notamment, ceci impose que le protocole IP soit implémenté en conformité avec l'ATM Forum.

Enfin, l'intégration de l'ATM dans le système d'exploitation implique la disponibilité des sources des pilotes associés aux cartes contrôleurs et impose le choix d'un fournisseur permettant une telle acquisition.

3.5 Intégration de l'ATM dans le système d'exploitation

L'ATM peut être utilisé de deux manières au sein d'un système d'exploitation : soit en remplacement d'un réseau local classique en offrant le protocole IP, soit en utilisation directe au niveau des circuits virtuels et des interfaces AALs. Par exemple, le sous-système de communication d'un micro-noyau (IPC) peut être étendu afin d'utiliser directement l'ATM (cf figure 4). Dans ces deux cas, les fonctions sont mises en œuvre au dessus d'un pilote approprié qui est ajouté au système de la machine. Généralement, les fournisseurs de cartes contrôleur proposent le binaire ou les sources du pilote, pour le système d'exploitation propriétaire de la machine auquel le contrôleur est destiné. Si on désire utiliser un autre système d'exploitation que le système propriétaire, il est alors nécessaire d'effectuer un portage des sources du pilote.

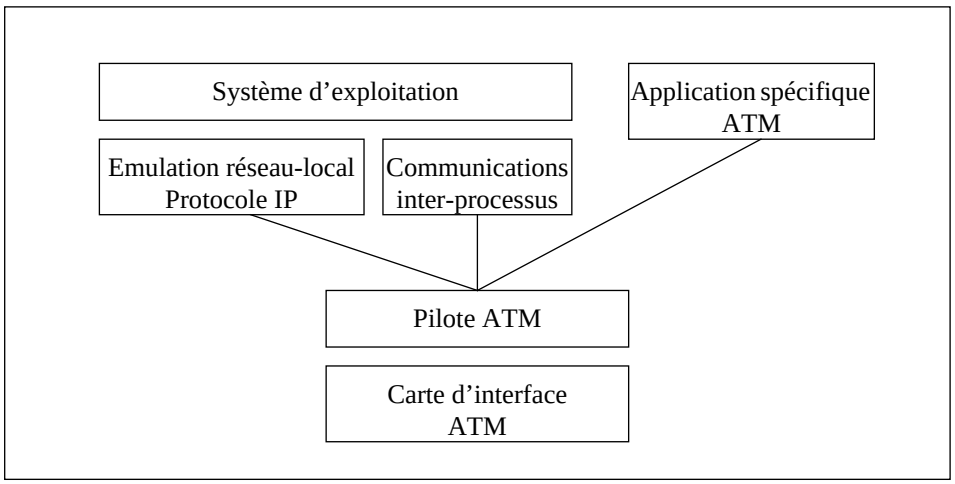

Figure 4 : Intégration de l'ATM dans un système d'exploitation

3.6 Contraintes imposées par l'ATM sur le choix des machines

L'ATM pose peu de contraintes sur le choix des machines, car des cartes d'interface sont aujourd'hui disponibles pour de nombreux bus d'entrées-sorties (ex : SBUS, EISA, MCA, TURBOchannel, SGI, VME). Il est de ce fait possible de connecter différentes sortes de machines au réseau, depuis des stations de travail mono ou bi-processeurs, jusqu'aux serveurs multi-processeurs de haute performance possédant de multiples bus d'entrées-sorties.

Néanmoins, nous avons vu dans le paragraphe 2.0.2 que pour des bus d'entrées-sorties tel que EISA ou SBUS, le débit réel de l'ATM était uniquement limité par la vitesse du processeur (SS10 : 56 Mbits/s, HP-PA 735 : 75 Mbits). Si on veut profiter du débit offert par l'ATM, il est de ce fait nécessaire de disposer de machines ayant des performances au moins équivalentes à ces machines. En se référant aux

3.3 Choix du débit des liaisons

Nous avons indiqué que la technologie ATM peut être mise en œuvre sur différents média. Deux types de liaisons sont à considérer : les liaisons stations-concentrateurs et les liaisons concentrateurs-concentrateurs.

Les liaisons stations-concentrateurs limitent potentiellement la performance des communications entre deux stations. Dans le paragraphe 2.0.1, nous avons vu qu'en utilisant des liaisons de débit supérieur ou égal à 100 Mbits/s (ATM et FDDI), il était possible réduire la latence d'un facteur au moins égal à 2. Par ailleurs, dans le cas de la liaison ATM à 140 Mbits/s, le temps de transfert sur le média représente un pourcentage minime (< 15%) du temps total de latence pour des RPCs avec relativement peu d'informations utiles (MaxArg : 1520+55 octets). Nous pouvons en conclure qu'une augmentation du débit des liaisons influerait assez peu sur la latence.

En ce qui concerne le débit utile, les résultats du paragraphe 2.0.2 montrent que le débit sur une liaison ATM à 100 Mbits/s dépend largement de l'architecture de la machine et de la vitesse du processeur. Pour les mêmes raisons que précédemment, il n'est pas évident que des liaisons de plus haut débit soient intéressantes compte tenu des stations de travail actuelles. Ceci explique que l'offre des constructeurs de contrôleurs ATM soit aujourd'hui limitée à débit maximum de 155 Mbits.

Le problème des liaisons concentrateurs-concentrateurs est différent car une liaison peut être saturée par les débits de plusieurs stations émettant simultanément. Deux stratégies, non exclusives, peuvent être utilisées en cas de congestion : régulation du trafic à la source et/ou augmentation de la capacité de la liaison concentrateur-concentrateur. L'accroissement la capacité des liaisons peut se faire (i) par réplication de la liaison (cf figure 3), dans ce cas, les protocoles de création de circuits virtuels doivent partager le trafic entre les différentes voies, (ii) par utilisation de liaisons de débit plus élevé de l'ordre de 600 Mbits/s. Ces liaisons ne sont pas aujourd'hui disponibles, mais sont annoncées pour l'an prochain (1995).

3.4 Choix des concentrateurs

Deux paramètres sont déterminants pour le choix d'un concentrateur : son architecture interne et sa capacité globale de commutation. L'architecture interne du concentrateur doit être modulaire, afin d'assurer l'extensibilité du nombre de liaisons et la migration vers de futurs types de liaisons.

La capacité globale de commutation permet d'évaluer la puissance d'un concentrateur. La capacité de commutation d'un concentrateur est calculée à partir du nombre et du débit des liaisons qui peuvent lui être raccordées sans qu'il y ait de perte de cellules ou délai d'attente. Par exemple, un concentrateur modulaire d'une capacité de 2,5 Gbits/s peut commuter 16 voies de 155Mbits/s ou encore 8 voies de 155 Mbits/s et 2 voies de 600 Mbits/s.

ajout direct de nouvelles stations à un concentrateur, soit par ajout d'un nouveau concentrateur au réseau.

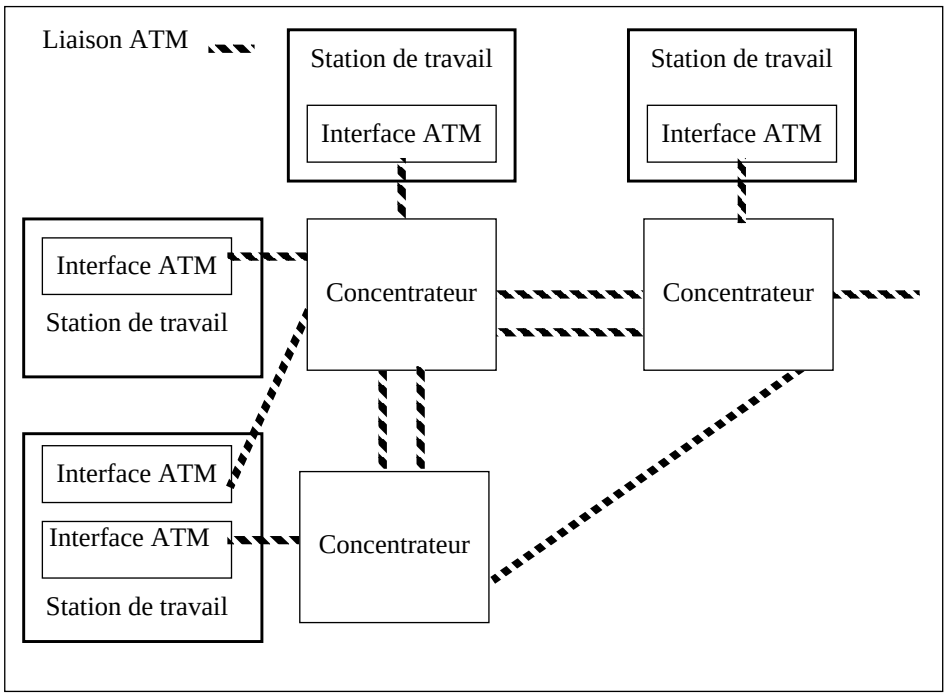

Figure 3 : Exemple de structure d'une plate-forme modulaire extensible

A partir d'un ensemble de concentrateurs, diverses topologies d'interconnexion entre les concentrateurs (ex: matrice, hypercube, étoile) sont possibles. Le choix d'une topologie et du nombre de stations connectées à un concentrateur est fortement dépendant du type d'application envisagée. Les concentrateurs disponibles aujourd'hui permettent de connecter jusqu'à 64 stations ou concentrateurs avec un temps de traversée minimum de 10 µs. Il est clair que si un temps de latence minimal est recherché, le nombre de concentrateurs à traverser pour faire dialoguer deux stations doit être minimisé et le nombre de stations connectées à un concentrateur maximisé.

Il est à noter que la défaillance d'un concentrateur entraîne un isolement des stations qui lui sont connectées. Si le but recherché est la disponibilité de la plate-forme, il peut être intéressant dans ce cas de borner le nombre de stations dépendant d'un concentrateur donné et ainsi réduire la perte potentielle de puissance de calcul. Par ailleurs, pour éviter l'isolation d'une station, il est possible de doubler la connexion au système en ajoutant une deuxième interface ATM vers un autre concentrateur (cf figure 3).

2. augmentation du nombre de machines sans introduction de contention réseau,

3. utilisation d'un réseau standardisé,

4. support du protocole Internet,

5. communications performantes,

6. débit et délai d'acheminement des données garantis.

3.1 Choix d'un réseau ATM

La contrainte la plus forte porte sur la garantie d'un débit ou d'un délai borné pour l'acheminement des données. D'après le paragraphe 3.1, seuls les réseaux ATM et 100 base VG le permettent. Il est à noter que ces deux réseaux possèdent la même topologie : liaisons point à point reliées par des concentrateurs. Ils sont donc extensibles par raccordement de stations ou de concentrateurs à un concentrateur existant.

Parmi ces deux réseaux, seul l'ATM est réellement disponible sur le marché. Indépendamment des contraintes imposées par le multimédia, l'ATM possède l'avantage d'être indépendant vis à vis des liaisons physiques et des débits. Il est de ce fait possible de mélanger des liaisons de débits différents sur la même plate-forme, ce qui peut s'avérer nécessaire pour des raisons d'hétérogénéité de matériel ou pour intégrer des liaisons plus performantes lorsqu'elles seront disponibles. Le choix de l'ATM permet, en plus, une évolution possible du médium dans le futur.

On peut noter que l'ATM satisfait également nos autres critères tels que le support du protocole IP et les communications performantes. D'après les évaluations du paragraphe 2.3, l'utilisation d'une liaison ATM 140 Mbits/s permet, pour une même machine, de réduire le temps de latence d'un RPC d'un facteur 2 à 3, par rapport à Ethernet. D'autre part, un facteur 7 peut être gagné, en débit, pour des transmissions fiables utilisant TCP.

3.2 Architecture générale de la plate-forme

L'architecture retenue pour notre plate-forme s'appuie sur un ensemble de concentrateurs reliés par des liaisons ATM (cf figure 3). A chaque concentrateur est attachée une grappe de stations de travail standard possédant une carte d'interface ATM. L'extension de la plate-forme peut être effectuée de deux manières : soit par

mise en œuvre utilisant un réseau ATM à 100 Mbits/s réalisé à partir de cartes FORE de la série 200 [Fore Systems 94].

Tableau 4 : Débit de TCP (en Mbits/s)

Buffer	PC 486/ 66Mhz Mach 3.0/BSD Ethernet	SUN SS 2-4/75 Ethernet	SUN SS 10/51 SunOs 4.1 Ethernet	SUN SS 2 ATM SBUS	SUN SS 10 ATM SBUS	HP-PA 900/735 ATM EISA
SPECint	32.4	21.8	65.2	21.8	65.2	80
4 Ko	2.85	5.5	7.27			
16 Ko	2.85	8.16	8.46	24.8	29.8	
32 Ko	2.85	8.16	8.46	35.6	42.3	
51 Ko	2.85	8.16	8.46	38.6	56.6	75.04

Les mesures du tableau 4 montrent qu'Ethernet peut être pratiquement saturé (85% du débit physique d'Ethernet). Cependant, ce résultat n'est obtenu que pour certaines machines (Sparc 10, Sparc 2, DPX) et ce uniquement pour des tailles de tampon appropriées ($\geq$ 16 Ko). Les raisons de l'inefficacité du PC sont les mêmes que celle décrites précédemment.

En utilisant une liaison ATM à 100 Mbits, on peut constater que le débit maximum obtenu est uniquement dépendant de la machine. Il est à noter que les bus EISA et SBUS ont des débits respectivement de 33 Mo/s et 50 Mo/s (débit soutenu), donc largement supérieurs à ceux nécessités par une liaison à 100 Mbits/s. De ce fait, on peut penser que la différence de débit constatée entre les machines vient de la vitesse d'exécution de TCP par le processeur, ce qui est à peu près confirmé par le rapport entre le débit et la performance mesurée en SPECint.

A titre de comparaison, pour la Paragon qui possède des liens à 1,6 Gbits (200 Mo/s), le débit maximal obtenu est de 75 Mo/s (sous OSF/1 version 1.1.1). Toutefois, sur une telle architecture, les communications sont supposées fiables et ne nécessitent donc pas l'utilisation de protocoles de transmission réalisant cette fonction, ce qui se traduit par un gain sur l'exécution du logiciel.

3 Proposition d'une architecture pour la plate-forme

Dans ce paragraphe, nous proposons une structure de plate-forme qui répond aux différents critères que nous rappelons :

1. possibilité d'ajout et de retrait de machines sans arrêt complet de la plate-forme,

des configurations non chargées. Les performances pour la configuration PC/Mach 2.5 sont tirées de [Maeda & Bershad 93].

Tableau 3 : Aller-retour au-dessus de TCP/IP

Taille du message (octets)	SUN SS 10/51 SunOs 4.1 Ethernet	SUN SS 2 SunOs 4.1 Ethernet	SUN SS 2 - DPX 20/630 SunOs 4.1 / AIX Ethernet	PC 486/ 66 Mhz Mach 3.0/ BSD Ethernet	PC 486/ 33 Mhz Mach 2.5 Ethernet
1	1160 µs	1200 µs	1200 µs	4400 µs	2080 µs
100	1250 µs	1600 µs	1600 µs	5800 µs	2690 µs
512	1790 µs	2400 µs	2400 µs	7500 µs	5450 µs
1024	2750 µs	3600 µs	3600 µs	9700 µs	8780 µs

Les résultats présentés dans le tableau 3 montrent que les performances des PCs sont très inférieures à celles obtenues pour les Sparcs et DPX. La première raison est que les contrôleurs Ethernet utilisés ne supportent pas le fonctionnement en mode DMA, les transferts sont alors effectués par le processeur. La seconde est la relative lenteur du bus d'entrée/sortie (ISA) qui ne supporte que le transfert d'un octet à la fois. Enfin, la version de système construite à partir d'un micro-noyau (Mach 3.0 + BSD) est moins performante que la version intégrée (Mach 2.5), car des recopies de tampons supplémentaires sont effectuées dans le noyau. Des solutions à ce dernier problème sont apportées dans [Maeda & Bershad 93].

La différence importante constatée entre les Sun SS2 et SS10 est notamment due à la vitesse du processeur et à l'architecture de la machine (débit mémoire).

2.3.2. Evaluation de transfert d'informations

L'évaluation du débit utile d'informations d'une connexion entre deux machines est complémentaire de l'évaluation de la latence. Nous avons vu précédemment que le débit de la liaison physique n'était pas forcément le facteur le plus important dans le temps de latence, étant donné le volume de code devant être exécuté à chaque appel. En revanche, lorsqu'un volume important de données est transmis, on peut a priori penser que le rapport du temps d'occupation du médium, sur le temps d'exécution des protocoles par le processeur, est grand et que le débit utile est seulement limité par celui de la liaison physique. Afin de vérifier l'hypothèse précédente, nous avons réalisé une évaluation de TCP/Ethernet au dessus des machines à notre disposition à l'IRISA. Nous avons également comparé ces résultats (cf tableau 4) avec une

les temps dus au logiciel seraient réduits par un facteur cinq dans le meilleur des cas (il est difficile d'estimer l'influence du cache). Ceci donnerait un temps d'exécution de MaxArg de 1500 µs pour Ethernet et de 307 µs pour l'ATM. En conséquence, le gain de l'ATM par rapport à Ethernet serait d'un rapport de 4,88.

Toutefois, les résultats concernant l'ATM sont à relativiser car une nouvelle génération de contrôleurs, plus performants mais plus complexes, est apparue depuis cette étude. Par exemple, dans la série FORE-200 [Fore Systems 94], la fragmentation des données et le ré-assemblage des cellules ATM sont effectués par le contrôleur réalisé à partir d'un processeur i960. On peut donc supposer que la transmission/réception de paquets de données contenant plusieurs cellules est maintenant plus efficace. En revanche, la latence minimale pour le transfert d'une cellule est potentiellement plus élevée que dans la série 100 du fait du temps de prise en compte et d'exécution de commandes par le processeur i960 du contrôleur.

A titre de comparaison entre les systèmes distribués et les multiprocesseurs faiblement couplés, nous avons réalisés les mêmes mesures sur la Paragon de l'Irisa en utilisant le système d'exploitation OSF/1. Pour cette machine, le débit des liens de communication entre processeurs est de 1,6 Gbits/s. Les résultats obtenus en utilisant la couche de communication NX sont de 330 µs pour Minus et de 479 µs pour Maxarg. Ces résultats qui se situent entre ceux obtenus avec la configuration DEC/réseau ATM, tiennent au fait que tout transfert sur le réseau de la Paragon nécessite 100 µs pour l'initialisation du chemin de données*.

Les résultats précédents représentent des minima obtenus avec des logiciels optimisés pour des systèmes d'exploitation (Ultrix, SunOs), des contrôleurs et des machines spécifiques. Ces résultats doivent donc être utilisés avec précaution. Une bonne conclusion est qu'il est très difficile d'estimer la performance d'un système de communication en se basant uniquement sur le type du réseau : Ethernet, FDDI ou ATM. L'architecture des machines, les contrôleurs, la vitesse du CPU et le logiciel sont également très importants.

Performances du protocole TCP/IP

Dans le paragraphe précédent, nous avons décrit l'obtention de temps de latence minimaux par optimisation de l'implémentation du RPC, notamment par utilisation de la couche datagramme. Toutefois, les systèmes d'exploitation et les applications distribués actuels reposent sur des protocoles largement diffusés dans la communauté Unix tels que UDP et TCP, ce dernier offrant la fiabilité des transferts. De ce fait, nous avons jugé intéressant d'estimer la performance de TCP au dessus de diverses machines et systèmes d'exploitation disponibles à l'IRISA. Nous avons évalué la latence minimum d'un aller et retour de données sur le réseau Ethernet dans

* Ces mesures ont été obtenues avec la version 1.1.1 d'OSF/1

média, le temps dépendant du matériel (traversée des contrôleurs, interruptions) et le temps dû au logiciel (gestion des appels, traversée des talons). Pour obtenir ces résultats, des hypothèses simplificatrices ont été effectuées sur la fiabilité des transferts et sur le nommage des stations (utilisation de la couche datagramme). De plus, la génération de talons et les recopies au sein du noyau ont été optimisées [Thekkath & Levy 93].

Tableau 2 : RPC optimisé (µs)

Procédure/ Taille des paramètres (octets)	Activité	Ethernet (DEC 5000/ 200)	Ethernet (SUN SS 1)	FDDI (DEC 5000/ 200)	ATM 140 Mbits (DEC 5000/ 200)
Minus 55 + 55	Média	117	117	9	6
	Matériel	102	143	209	50
	Logiciel	121	236	200	114
	Total	340	496	380	170
MaxArg 1520 + 55	Média	1278	1278	127	91
	Matériel	103	152	290	125
	Logiciel	689	567	276	459
	Total	2070	1997	693	675

Ces résultats montrent que pour Ethernet, le surcoût minimum du RPC est d'environ 30% du temps d'aller-retour matériel (comparaison avec le tableau 1). En revanche, pour les réseaux haut débit le surcoût du RPC est nettement plus élevé (130% pour l'ATM). Ceci s'explique par le fait que le temps de transfert sur le médium (dépendant du débit physique) entre pour une part minime dans le temps total et que la performance du RPC est dépendante de la vitesse du processeur, de l'architecture de la machine et du temps de traversée du contrôleur.

L'ATM permet une réduction de la latence, par rapport à Ethernet, d'un facteur deux pour la procédure Minus et d'un facteur trois pour la procédure MaxArg. Pour cette dernière procédure, le logiciel représente 33% du temps de latence pour Ethernet et 68% pour l'ATM. En utilisant le SPECint comme unité de mesure de la vitesse des processeurs*, on peut essayer d'extrapoler ces résultats pour des machines actuelles plus performantes que le DEC 5000/200 (19.5 SPECint), par exemple un DEC ALPHA AXP 600 (114 SPECint). Avec un AXP 600, on peut estimer que

* Un certain nombre de benchmarks actuels tels que Specint possèdent le désavantage qu'ils s'exécutent entièrement dans le cache du processeur et ne mesurent pas le temps d'accès à la mémoire.

Levy 93]. Nous donnons ici quelques résultats de cette dernière étude. Elle concerne différents matériels et réseaux en séparant les temps dus au matériel de ceux dus aux différentes couches logicielles.

Latence d'échange de messages

Le tableau 1 donne les temps minimaux d'échanges de messages d'"aller et retour" pour différentes machines et réseaux. Le temps total est la somme des temps suivants : temps sur le médium, traversée du contrôleur, transfert des données, gestion des interruptions. Ces résultats ont été obtenus sur des réseaux non chargés et, dans le cas de l'ATM, sur une liaison unique bi-point. La contention d'accès au média n'est donc pas mesurée. La première constatation est que pour des petits messages (cas de synchronisation), le débit du réseau n'influe pas sur les résultats. Ceux-ci dépendent en revanche des boîtiers contrôleurs. Le meilleur temps est obtenu avec l'ATM grâce à la simplicité du contrôleur (FORE série 100) qui permet d'obtenir un temps d'accès plus faible au médium que pour FDDI, alors que ces deux réseaux utilisent le même type de médium (circuits TAXI, fibre optique).

Tableau 1 : Performances du matériel (aller + retour) en µs

Taille du paquet (octets)	Ethernet (DEC 5000/ 200)		Ethernet (SparcStation 1)		FDDI (DEC 5000/200)		ATM 140 Mbits (DEC 5000/ 200)	
	Média	Total	Média	Total	Média	Total	Média	Total
60/60	115	253	115	263	13	263	6	73
1514/ 1514	2442	3137	2442	2611	245	894	176	746

Lorsque la taille des messages échangés s'accroît, la situation est plus complexe, elle dépend des contrôleurs et de la nature du réseau. On peut toutefois noter que les temps de latence obtenus avec les deux réseaux à haut débit sont en moyenne supérieurs de quatre fois à ceux d'Ethernet. Les performances réelles de l'ATM sont nettement inférieures à ce que le débit de 140 Mbits laisserait espérer. Ceci est dû à la nécessité de ré-assembler les cellules par le processeur de la station pour cette génération de carte. Ceci n'est plus vrai pour les contrôleurs actuels tels que ceux de la gamme FORE-série 200. La différence entre les deux implémentations d'Ethernet s'explique aussi par le fait que les transferts sont effectués par la DMA sur le SparcStation 1 en recouvrement avec l'arrivée de la trame, alors que sur le DEC 5000, le transfert est effectué par le processeur une fois la trame arrivée dans le contrôleur.

Latence de RPC élémentaires

Le tableau 2 présente les temps d'exécution d'appels de deux procédures à distance (Minus et MaxArg) comportant peu de paramètres et qui sont de ce fait représentatives d'une synchronisation. Nous avons différencié le temps d'occupation du

d'information en transit. De ce fait, la latence peut être mesurée en secondes, de manière indépendante du volume d'information.

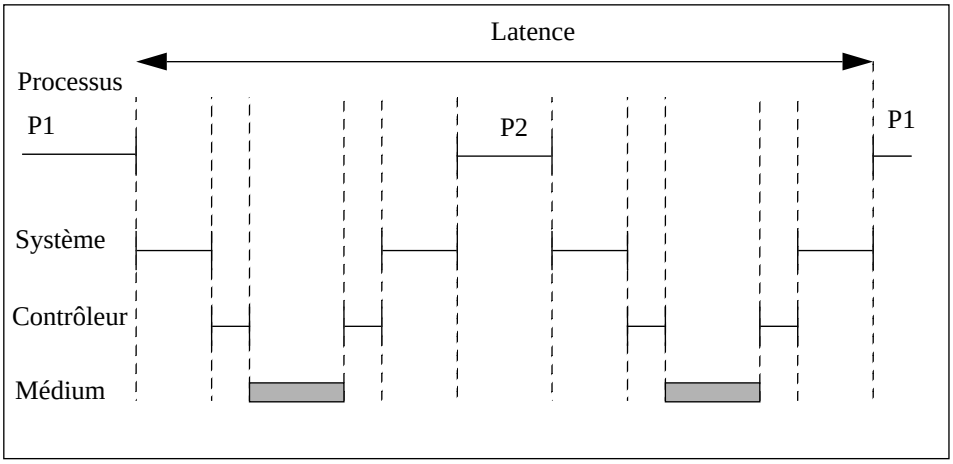

Figure 2 : Synchronisation entre deux processus

Dans le cas d'un transfert d'un volume d'informations important, les temps de traversée du système et des contrôleurs deviennent a priori petits, voire négligeables, devant le temps d'émission sur le médium. On peut donc penser que dans ce cas, le temps de transfert des informations est surtout limité par le débit physique du médium.

2.3.1. Evaluation de la latence

La forme la plus élémentaire de synchronisation est l'échange de deux messages (ou paquets) entre deux machines. Nous distinguons deux niveaux dans la mise en œuvre d'un protocole de synchronisation : l'"aller-retour" de trames réseaux (niveau datagramme) et l'appel de procédure à distance (RPC). L'"aller-retour" de trames met en jeu uniquement le matériel et les pilotes d'entrées-sorties. De ce fait, la mesure des performance d'un "aller-retour" permet d'évaluer la performance de l'architecture de la machine et du réseau.

Dans un système d'exploitation réel, l'"aller-retour" de trames n'est pas utilisable par les processus pour des raisons de protection. Généralement, la synchronisation est effectuée par appel de procédure à distance. Par rapport à un échange de trames "aller-retour", l'implémentation du RPC nécessite en plus des changements de mode de protection entre l'utilisateur et le système et la traversée de talons qui réalisent l'empaquetage et le dépaquetage des paramètres. De ce fait, un RPC nécessite l'exécution d'un volume de code non négligeable, la vitesse du processeur devient un paramètre non négligeable du temps total d'exécution d'un RPC.

Les temps d'exécution minimaux d'"aller-retour" de trames, ou de RPC ont fait l'objet de plusieurs évaluations [Johnson & Zwaenepoel 91] [Thekkath &

La couche de signalisation Q.93B est particulièrement importante ; c'est elle qui permet l'interconnexion de matériels de constructeurs différents. Il est à noter que la mise en œuvre d'IP au dessus de l'ATM est en cours de standardisation (nommage ATM <-> IP). Certaines mises en œuvre sont déjà disponibles mais reposent sur des protocoles propriétaires (ex : SPANS pour FORE). De même, les protocoles de gestion de la congestion sont également en cours de normalisation. A l'heure actuelle, plusieurs liaisons physiques ont été normalisées par l'ATM Forum :

- Fibre optique : 100 Mbits TAXI
- DS3 : 44.736 Mbits
- E3 : 34 Mbits
- E4 : 139 Mbits
- Sonet-SDH STS-3c : 155 Mbits (paire torsadée, fibre optique)

Par ailleurs d'autres liaisons physiques sont en cours de normalisation ou en cours de développement :

- 25 Mbits/s IBM (paire torsadée)
- 140 Mbits FORE
- futur -> 600 Mbits, radio ?

2.3 Influence du réseau sur les performances de la plate-forme

En général, l'estimation des performances d'un système est toujours une tâche difficile, surtout dans le cas d'un système distribué où les communications influent sur l'ensemble des performances du système. Nous pouvons distinguer deux formes de communications : le transfert d'informations et la synchronisation. Dans le premier cas, la performance se mesure en nombre de Kilo ou Méga-octets transmis par seconde (débit). Dans le second cas, la mesure est effectuée en seconde ; la durée d'exécution d'une synchronisation est généralement appelée temps de latence. Il est à noter que la mise en œuvre de manière optimisée de ces deux formes de communication est parfois contradictoire.

Le temps d'exécution d'une communication entre deux processus se décompose en un temps de transfert sur le média, un temps d'initialisation et de traversée des contrôleurs d'entrées-sorties et un temps de traversée des couches du système d'exploitation dépendant de la complexité des protocoles utilisés (cf figure 2). Une synchronisation est caractérisée par un transfert de peu d'informations utiles. Dans ce cas, le temps de transfert sur le médium n'est pas forcément le paramètre le plus important, en regard des temps de traversée des couches systèmes et de gestion des contrôleurs d'entrées-sorties qui ont des valeurs minimales indépendantes du volume

d'une taille fixe de paquet (trame) de 53 octets (5 entête/ 48 données) et sur son indépendance vis à vis des liaisons physiques. De ce fait, l'ATM est déjà disponible sur plusieurs média et peut être utilisé pour construire des réseaux locaux ou publics. Pour cette même raison, la construction de passerelles "réseaux locaux-réseaux publics" est relativement simple. Dans le futur, l'augmentation du débit des liaisons pourra être effectuée sans remettre en cause l'ensemble d'un système existant.

Ce réseau est extensible via l'interconnexion de concentrateurs. Le routage des paquets est effectué au moyen de circuits virtuels. Un des arguments avancés par les promoteurs du réseau ATM est que la taille fixe des cellules et l'utilisation de circuits virtuels permettent d'optimiser la construction des concentrateurs afin de réduire leur temps de traversée (temps minimum de 10 µs pour le concentrateur FORE). Comme les liaisons sont de type point à point, il n'y pas de contention d'accès au média, ce qui rend l'ATM utilisable pour les applications multimédia. En revanche, il peut y avoir perte de paquets au niveau d'un concentrateur si le débit total de plusieurs entrées émettant vers une même sortie est supérieur au débit de celle-ci, même si les files d'attente internes aux concentrateurs permettent d'amortir les pointes de trafic.

L'ATM est actuellement normalisé par deux organisations : le CCITT et l'ATM Forum*. La dernière version promue par l'ATM Forum, "UNI 3.0", spécifie un certain nombre de couches dont celle de signalisation (Q.93B), la segmentation en cellules (AAL) et l'interface physique. Plusieurs couches AAL ont été spécifiées suivant que l'on désire transmettre du son, de la vidéo ou des données (cf figure 1).

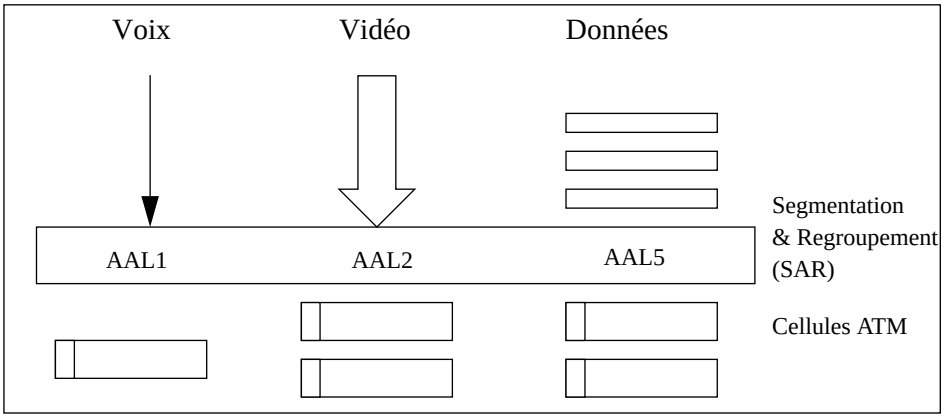

Figure 1 : Couche de segmentation et ré-assemblage

* L'ATM Forum est une association regroupant des constructeurs et des utilisateurs ayant pour but de promouvoir spécifiquement la technologie ATM et de proposer des standards.

2.2.1. Ethernet standard (10 Mbits)

Ethernet est le premier réseau local largement utilisé dans un environnement de stations de travail. Les différentes machines sont connectées à un médium unique. L'accès est de type CSMA/CD. Lorsqu'une station émet une trame, une collision peut éventuellement se produire avec une trame émise par une autre machine ; dans ce cas, il y a ré-émission au bout d'un temps aléatoire. L'extensibilité des réseaux de type CSMA/CD est limitée par une augmentation de la contention due à l'émission simultanée de trames par différentes machines. Une solution au problème de contention est de segmenter le réseau et de placer des répéteurs filtrants entre les segments afin d'isoler les trafics locaux. La méthode d'accès CSMA/CD pose également des problèmes pour les applications multimédia, car la ré-émission aléatoire en cas de collision ne permet pas de respecter des contraintes temporelles sur l'acheminement des données.

2.2.2. Ethernet 100 base T

Ce réseau est une évolution de la version de base d'Ethernet 10 Mbits à un débit de 100 Mbits. L'accès au médium reste de type CSMA/CD. Les deux versions peuvent être mixées sur un même segment. Ce réseau est en cours de développement et de normalisation.

2.2.3. Ethernet 100 base VG

Ce réseau est également une évolution d'Ethernet 10 Mbits à un débit de 100 Mbits. La différence essentielle avec ce dernier est que l'accès à un câble unique (CSMA/CD) est remplacé par des liaisons point à point reliées par des concentrateurs. De ce fait, il n'y a plus de contention d'accès au média. Il est donc possible de respecter des contraintes temporelles, ce qui le rend utilisable pour des applications multimédia. Ce type de réseau est en cours de développement ; il entre en concurrence commerciale avec 100 base T et l'ATM. Cependant, on ne connaît pas actuellement sa pérennité.

2.2.4. FDDI

Ce réseau est le premier réseau dit "haut débit" (100 Mbits) à avoir été introduit. Il possède une topologie en boucle dont l'accès est contrôlé par un jeton. La boucle peut éventuellement être doublée afin de tolérer les défaillances et permettre l'extensibilité sans interruption de service. Le temps maximal de garde du jeton par une station est de 4 ms. De ce fait, bien que le débit de FDDI soit 10 fois supérieur à Ethernet, ce réseau possède un temps maximal d'accès au média élevé. Ceci le rend difficilement utilisable pour des applications multimédia ou pour la synchronisation des applications distribuées.

2.2.5. ATM (Asynchronous Transfert Mode)

Un réseau ATM est constitué de liaisons de type point à point reliées par des concentrateurs [Boisseau *et al.* 94]. L'originalité de ce réseau repose sur l'utilisation

- La **performance des communications** influe sur la performance globale des systèmes distribués de recherche, car ceux-ci sont caractérisés par la répartition de leurs fonctions sur les différentes machines du système. Nous définissons la performance des communications entre deux processus résidants sur deux machines différentes, de deux manières : (i) par le débit utile d'informations et (ii) par la latence (temps de communication minimal). Dans le paragraphe 2.3, nous étudions les limites qu'il est possible d'atteindre pour le débit et la latence en fonctions du type de réseau et des machines utilisées.
- Les applications **multimédia** supposent de façon générale des communications performantes. Par rapport aux applications scientifiques traditionnelles, le support des applications multimédia impose, en plus, des contraintes temporelles sur les données, telles qu'un débit et un délai d'acheminement bornés.

La liste suivante résume les critères que nous avons retenu pour le choix du réseau :

1. possibilité d'ajout et de retrait de machines sans arrêt complet de la plate-forme,
2. augmentation du nombre de machines sans introduction de contention réseau,
3. utilisation d'un réseau standardisé,
4. support du protocole Internet,
5. communications performantes,
6. débit et délai d'acheminement des données garantis.

Nous examinons maintenant les différents réseaux locaux existants afin de déterminer dans quelle mesure ils répondent à ces critères.

2.2 Présentation des principaux réseaux locaux en environnement de station de travail

Depuis l'introduction d'Ethernet, il y a une quinzaine d'années, plusieurs réseaux ont fait leur apparition sur le marché (Ethernet, FDDI, ATM, 100 base T, 100 base VG). Nous les passons brièvement en revue en soulignant leur principales caractéristiques. Nous nous sommes volontairement limités aux réseaux standardisés ou en cours de standardisation.

2 Aspects relatifs au choix d'un réseau

Le réseau représente en quelque sorte l'épine dorsale d'un système distribué. Sa structure influe donc sur caractéristiques et les performances potentielles de l'ensemble du système. Dans la suite de cette partie, nous présentons successivement les critères retenus pour le choix du réseau, les caractéristiques des principaux réseaux locaux existants et l'impact du réseau sur les performances du système.

2.1 Critères de choix du réseau

La construction d'une plate-forme de calcul distribuée doit satisfaire les besoins des futures applications (scientifiques, coopératives, multimédia). Ce sont principalement l'extensibilité, l'hétérogénéité, les communications inter-machines performantes, le support des applications multimédia et la compatibilité avec les systèmes Unix interconnectés existants. Nous précisons maintenant ces besoins afin d'en déduire les critères de choix du réseau :

- L'**extensibilité** caractérise la possibilité d'accroître la performance globale du système par ajout de machines. L'extensibilité découle pour une part de la topologie du réseau ; l'augmentation du nombre de machines doit pouvoir se faire sans introduire de points de contention. De plus, il est important que l'ajout ou le retrait de machines d'un système soient réalisés sans rendre indisponible la plate-forme informatique aux utilisateurs. Par exemple, le raccordement d'une station à un réseau Ethernet s'effectue par une "piqûre" dans le câble réseau du connecteur d'interface de la station, et de ce fait ne coupe pas le câble. En revanche, l'insertion d'une station dans un réseau en boucle nécessite une ouverture de celle-ci ; il en résulte une coupure temporaire des liaisons.
- L'**hétérogénéité** caractérise l'interconnexion au sein d'un même système de machines ayant des architectures différentes et dont les processeurs peuvent posséder des formats de représentation des données incompatibles. L'hétérogénéité résulte soit de l'utilisation de machines spécialisées, soit de l'évolution rapide de la technologie qui se traduit par la présence de générations de machines différentes au sein d'une même plate-forme. Le traitement de l'hétérogénéité est essentiellement résolu par le système d'exploitation. Cependant, au niveau physique, l'intégration de machines hétérogènes nécessite l'utilisation d'un réseau s'appuyant sur des standards reconnus et pour lesquels différents constructeurs développent des cartes d'interface.
- La **compatibilité avec les systèmes Unix interconnectés** est nécessaire pour utiliser la plate-forme non seulement pour les travaux de recherche, mais aussi pour les travaux quotidiens tels que l'édition de documents et la production de programmes. Les systèmes Unix sont aujourd'hui interconnectés par des services distribués tel que NFS, FTP, X-windows qui reposent sur le protocole Internet (IP). Il est donc indispensable que le réseau supporte ce protocole.

D'autre part, l'évolution du matériel doit être effectuée tout en préservant les logiciels développés précédemment. Jusqu'à présent la réutilisation de logiciels s'est avérée difficile, voire impossible. La mise en place de la plate-forme, perçue comme un dénominateur commun de développements logiciels, serait une réponse à ce problème de la pérennité des développements. Ceci suppose d'intégrer l'hétérogénéité dans nos choix tant au niveau du matériel que du système opératoire.

1.2 Contenu du document

Notre objectif est de construire dans les deux ans à venir une plate-forme informatique distribuée composée d'une trentaine de machines. Cette plate-forme repose sur trois éléments : un réseau, des stations de travail et un (ou plusieurs) systèmes d'exploitation. Le choix de ces trois éléments doit se faire en fonction de nos besoins futurs, du moins ceux que nous pouvons estimer actuellement. Par exemple, le système devra pouvoir servir de base de développement de machines langages ou supporter des extensions noyaux. De même, il devra être aisément portable afin d'évoluer avec les nouvelles générations de machines.

Ce document décrit les contraintes de développement de cette plate-forme ainsi que nos choix pour son architecture générale et ses éléments constitutifs. Le paragraphe 2 est consacré au réseau et à l'influence de celui-ci sur les performances de la plate-forme. À partir de cette analyse, nous proposons dans le paragraphe 3, une architecture de plate-forme basée sur le réseau ATM. Le paragraphe 4 présente nos besoins en matière de système d'exploitation et les offres industrielles reposant sur la technologie micro-noyau. Cette étude nous amené à choisir l'offre de Chorus Système tant pour le micro-noyau que pour le système Unix. En fonction des contraintes dues au réseau ATM et au micro-noyau Chorus, nous décrivons dans le paragraphe 5, le choix de stations de travail à base de PC/Pentium. Enfin, nous concluons dans le paragraphe 6 par la synthèse récapitulative de nos choix.

1 Introduction

L'évolution de la technologie des microprocesseurs et des systèmes de communication laisse présager, dans un avenir proche, l'émergence de nouvelles architectures distribuées extensibles dont la puissance croît en fonction du nombre d'éléments processeurs. Leurs champs d'application relèvent aussi bien des applications distribuées traditionnelles de type "client-serveur" que des applications parallèles nécessitant une grande puissance de calcul (e.g. réalité virtuelle, applications coopératives, multimédia, bases de donnée). Toutefois, une différence essentielle avec les applications dédiées aux multiprocesseurs classiques (i.e. centralisés), est la distribution géographique des entités manipulées au sein d'une application, avec tous les problèmes sous-jacents que ceci entraîne (protection, disponibilité, temps d'accès). Bien entendu, la prise en compte de cette distribution ne doit pas se faire au détriment de la puissance attendue. De ce fait, il est nécessaire de disposer d'une plate-forme d'expérimentation qui soit utilisable à la fois pour les recherches en systèmes distribués et pour le développement de nouvelles applications parallèles.

1.1 Buts de la plate-forme

Les besoins posés par la recherche en systèmes et applications distribués sont de plusieurs types : nécessité de disposer d'un outil de taille "réelle", préservation des développements, suivi de l'évolution technologique.

Un problème essentiel dans la conception d'un système informatique est l'analyse et l'optimisation de son comportement en utilisation réelle. S'il est relativement facile d'étudier un système ou une application sur une machine unique, le problème est beaucoup plus difficile pour les systèmes distribués. En effet, beaucoup de prototypes qui se comportent de manière acceptable pour une architecture limitée à la dizaine de machines, voient leurs performances se dégrader lorsque le nombre de machines dépasse cette limite. Pour étudier le comportement des systèmes ou d'applications distribués en grandeur réelle, il est donc nécessaire de disposer dès la construction du prototype d'une plate-forme d'expérimentation de plusieurs dizaines de machines.

La construction d'une telle plate-forme nécessite aussi de prendre en compte des problèmes tels que le suivi de l'évolution technologique et la préservation des développements. D'une part, la rapidité d'évolution des calculateurs entraîne une durée de vie relativement courte, environ deux ans, des gammes des constructeurs. La présentation de résultats de recherche sur des machines obsolètes est toujours malaisée et il est difficile d'extrapoler des résultats sur des nouvelles générations de machines. Il est donc nécessaire d'intégrer l'évolution du matériel dès la conception de la plate-forme. Cette évolution doit être graduelle, car pour des raisons évidentes de coût il n'est pas envisageable de remplacer plusieurs dizaines de machines simultanément.

An Architecture for Scalable Distributed Systems Experimentation

Abstract: The evolution of microprocessor technology and communication networks leads to new scalable distributed architectures those processing power increase with the number of processors. Purposes of these architecture are traditional “client-server” distributed applications or parallel number crunching ones (e.g., virtual reality, co-operative working, multimedia application, database). Moreover, research in distributed systems and distributed applications introduces problems such as the need of having a “real size” framework, preserving existing developments and chasing technology evolution. This document presents the design choices for a platform oriented to distributed experimentations which both meets the research needs in distributed architectures and systems, and allows the development of new distributed and/or parallel applications. We first give the technical requirements for the building of our platform. Then, we detail the global platform architecture and our choice for each of its components (e.g., LAN, operating system, workstations).

Key-words: large scale distributed systems, micro-kernel, ATM, multimedia application

Proposition d'une architecture pour l'expérimentation de systèmes distribués extensibles

Michel Banâtre et Gilles Muller*

Programme 1 - Architectures parallèles, bases de données, réseaux et systèmes distribués

Projet SOLIDOR

Rapport de recherche n°2506

Octobre 1994

33 pages

Résumé : L'évolution de la technologie des microprocesseurs et des systèmes de communication laisse présager, dans un avenir proche, l'émergence de nouvelles architectures distribuées extensibles dont la puissance croît en fonction du nombre d'éléments processeurs. Leurs champs d'application relèvent aussi bien des applications distribuées traditionnelles de type "client-serveur" que des applications parallèles nécessitant une grande puissance de calcul (e.g., réalité virtuelle, applications coopératives, multimédia, bases de donnée). Par ailleurs, la recherche en systèmes et applications distribués pose des problèmes tels que la nécessité de disposer d'un outil de taille "réelle", la préservation des développements et le suivi de l'évolution technologique. Ce document présente les choix de conception d'une plate-forme d'expérimentation qui intègre ces différents besoins et soit utilisable à la fois pour valider nos recherches en architectures et systèmes distribués et pour le développement de nouvelles applications parallèles et/ou réparties. Après avoir donné nos contraintes de développement, nous présentons l'architecture générale de la plate-forme ainsi que nos choix pour ses éléments constitutifs (e.g., réseau, système d'exploitation, station de travail).

Mots-clé : systèmes distribués, micro-noyau, ATM, multimedia.

(Abstract: pto)

* Email : Michel.Banatre@irisa.fr | Gilles.Muller@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex (France)
Téléphone : (33) 99 84 71 00 - Télécopie : (33) 99 38 38 32

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE

Proposition d'une architecture pour l'expérimentation de systèmes distribués extensibles

Michel Banâtre et Gilles Muller

N° 2506

Octobre 1994

PROGRAMME 1

ISSN 0249-6399